



INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA
CAMPUS MONTEIRO
DIRETORIA DE ENSINO E DESENVOLVIMENTO
CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE
SISTEMAS

DAYVID YAGGO PEREIRA SOUSA

RELATÓRIO DE ESTÁGIO PHOEBUS

MONTEIRO

2023

DAYVID YAGGO PEREIRA SOUSA

RELATÓRIO DE ESTÁGIO PHOEBUS

Relatório de Estágio apresentado à
Coordenação de Estágio do Instituto
Federal de Educação, Ciência e
Tecnologia da Paraíba, Campus Monteiro,
como requisito parcial para conclusão do
Curso Superior Tecnólogo em
Análise e Desenvolvimento de Sistemas

Orientador: **Prof. Msc. Renata França de
Pontes**

MONTEIRO

2023

DAYVID YAGGO PEREIRA SOUSA
RELATÓRIO DE ESTÁGIO PHOEBUS

Relatório de Estágio apresentado à
Coordenação de Estágio do Instituto
Federal de Educação, Ciência e
Tecnologia da Paraíba, Campus Monteiro,
como requisito parcial para conclusão do
Curso Superior Tecnólogo em
Análise e Desenvolvimento de Sistemas

Aprovado em 08 de Março de 2023.

BANCA EXAMINADORA

Documento assinado digitalmente
 **RENATA FRANCA DE PONTES**
Data: 30/04/2025 11:55:30-0300
Verifique em <https://validar.iti.gov.br>

Prof. Ma Renata França de Pontes (Orientadora - IFPB)

Documento assinado digitalmente
 **CLEYTON CAETANO DE SOUZA**
Data: 30/04/2025 10:46:19-0300
Verifique em <https://validar.iti.gov.br>

Prof. Dr. Cleyton Caetano de Souza (Examinador - IFPB)

Documento assinado digitalmente
 **ELENILSON VIEIRA DA SILVA FILHO**
Data: 30/04/2025 11:43:56-0300
Verifique em <https://validar.iti.gov.br>

Prof. Me. Elenilson Vieira da Silva Filho (Examinador - IFPB)

RESUMO

Neste relatório, estão descritas as atividades desenvolvidas durante a atuação profissional do discente na área de Análise e Desenvolvimento de Sistemas no período correspondente como funcionário da empresa Phoebus. Elas estão descritas de maneira abrangente, para que não haja a quebra de sigilo com a empresa. As atividades aqui descritas mostram novas tecnologias aprendidas, continuação do aprendizado desenvolvido no curso, como Java e Angular, dificuldades encontradas e o aumento nos desafios, além do aumento da confiança da empresa para solicitar atividades mais robustas.

Palavras-chave: Java; estágio; angular

ABSTRACT

In this report, the activities developed during the professional performance of the student in the area of Systems Analysis and Development in the corresponding period as an employee of the company Phoebus are described. They are described in a comprehensive way, so that there is no breach of confidentiality with the company. The activities described here show new technologies learned, continuation of learning developed in the course, such as Java and Angular, difficulties encountered and the increase in challenges, in addition to increasing the confidence of the company to request more robust activities.

Key-words: Java; internship; angular

LISTA DE FIGURAS

Figura 01 - Modelo de padrão de projeto DTO para a entidade Book.....	11
Figura 02 - Pasta v1 com o controller de livro.....	11
Figura 03 - Pasta com os serviços da entidade livro.....	12
Figura 04 - Entidade BookCategory.....	12
Figura 05 - Entidade Client.....	13
Figura 06 - Entidade Purchase.....	13
Figura 07 - Pasta com as exceções da aplicação.....	14
Figura 08 - Pasta utilizada pelo flyway	14
Figura 09 - Testes unitários da entidade Book.....	15
Figura 10 - Projetos em separado, com o server.....	16
Figura 11 - Tela do portal para controle do servidor de email.....	19
Figura 12 - Tela do portal com as novas colunas.....	21

SUMÁRIO

RESUMO.....	3
ABSTRACT.....	4
LISTA DE FIGURAS.....	5
SUMÁRIO.....	6
1 INTRODUÇÃO.....	7
2 OBJETIVOS.....	8
2.1 Objetivo geral.....	8
2.2 Objetivos específicos.....	8
3 RELATO DE ESTÁGIO.....	9
3.1 Identificação da empresa.....	9
3.1.1 Local do estágio e aspectos físicos da empresa.....	9
3.1.2 Dados referentes ao estágio.....	9
3.1.3. Perfil da empresa.....	9
3.2 Atividades desenvolvidas.....	10
3.2.1 Projeto de capacitação.....	10
3.2.2 Desenvolvimento de funcionalidades do PayStore.....	16
3.2.3 Projeto PayStore - Release 2021.02.....	17
3.2.4 Projeto PayStore - Release 2021.03.....	18
3.2.5 Projeto PayStore - Release 2021.04.....	19
3.2.6 Projeto PayStore - Release 2021.05.....	20
3.2.7 Projeto PayStore - Release 2021.06.....	20
3.2.8 Projeto PayStore - Release 2021.07.....	21
3.2.9 Projeto PayStore - Release 2021.08.....	21
3.2.10 Projeto PayStore - Release 2021.09.....	22
3.2.11 Projeto PayStore - Release 2021.10.....	22
3.2.12 Projeto PayStore - Release 2021.11.....	23
3.2.13 Projeto PayStore - Release 2021.13.....	23
3.2.15 Projeto PayStore - Release 2021.14.....	24
3.2.16 Projeto PayStore - Release 2021.15.....	24
3.2.18 Projeto PayStore - Release 2022.01.....	24
3.2.19 Projeto PayStore - Release 2022.02.....	24
4 CONSIDERAÇÕES FINAIS.....	25
REFERÊNCIAS.....	26
GLOSSÁRIO.....	28

1 INTRODUÇÃO

O trabalho relatado aqui é destinado à conclusão do Curso Superior em Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia da Paraíba - Campus Monteiro. Nele, são descritas as atividades desenvolvidas no período de trabalho na empresa Phoebus.

A empresa Phoebus é uma multinacional que foi pioneira no ramo de terceirização do sistema das máquinas de cartão de crédito, além de desenvolver um sistema que faz o controle das mesmas (PHOEBUS, 2022).

Um ponto que os produtos dão suporte é a integração com diversos adquirentes. Adquirentes são os responsáveis por se comunicar com bandeiras para as autorizar os facilitadores. Facilitadores são os que vendem as máquinas de cartão e negociam as taxas. Assim, eles podem buscar adquirentes com as melhores taxas, o que acarretaria em deixar seu produto mais atrativo. O facilitador também pode negociar com outros adquirentes que aceitem outros tipos de bandeiras, deixando para o lojista uma máquina que aceite as mais diversas bandeiras.

Na empresa, são usadas tecnologias como: Angular (ANGULAR, 2022), Spring (SPRING, 2022), BD Oracle (ORACLE, 2022), BD Mongo (MONGO, 2022), tecnologias essas já conhecidas pelo estagiário e que serão mais detalhadas ao longo do relatório.

2 OBJETIVOS

2.1 Objetivo geral

O trabalho teve por objetivo o desenvolvimento full-stack do aluno nos projetos da empresa, de tal forma que acarretassem em uma melhora profissional e que gerassem a empresa uma nova visão para novos produtos e a continuação e a melhora dos produtos já existentes.

2.2 Objetivos específicos

- Encontrar e resolver bugs em toda a linha de produtos.
- Mostrar pontos de melhoria em pontos já existentes e em implementações novas.
- Aprender tecnologias usadas na empresa.
- Acompanhar reuniões com clientes onde são mostradas as novas implementações que o mesmo solicitou.

3 RELATO DE ESTÁGIO

3.1 Identificação da empresa

3.1.1 Local do estágio e aspectos físicos da empresa

A empresa Phoebus tem como sede principal a cidade de João Pessoa, no DCT (Duo Corporate Towers) Torre Norte - 29º e 30º Andar



3.1.2 Dados referentes ao estágio

No dia 01/03/2021, foi dado início ao contrato com carteira assinada com carga horária de 40 horas semanais vigente até hoje.

3.1.3. Perfil da empresa

A empresa segue no setor de maquinetas de cartão, capturando e processando as transações. Ela surgiu de dentro de uma faculdade, em 1997, com a ideia de três alunos e um professor e foi a primeira no seu setor a conseguir um CMMI nível 2, além de possuir MPS.BR nível F e equipe de gerenciamento de projetos certificada pelo PMI (Project Management Institute).

3.2 Atividades desenvolvidas

As atividades descritas e desenvolvidas no presente relatório se referem ao período em que o discente atuou como desenvolvedor *full stack* na empresa Phoebus.

3.2.1 Projeto de capacitação

A primeira fase na empresa foi destinada a capacitação do aluno, para que pudesse desempenhar, posteriormente, as atividades definidas, da melhor forma possível. Essa capacitação foi dividida em duas etapas: aprender o padrão organizacional da empresa e tecnologias utilizadas; aprender sobre microsserviços.

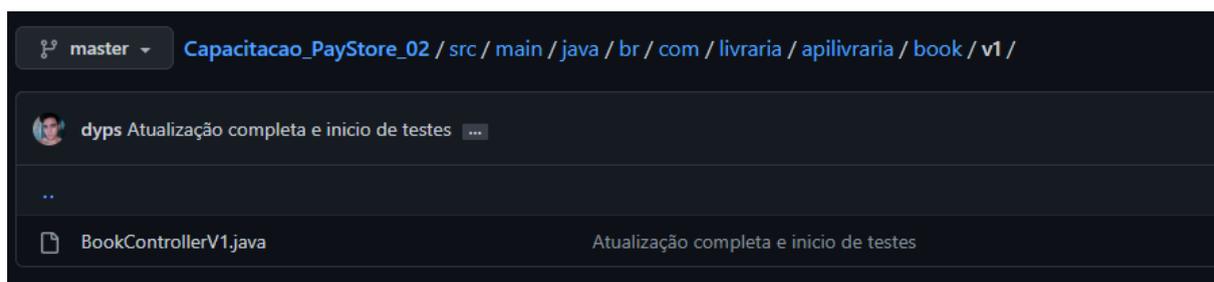
Etapa 1: Desenvolver uma API Spring (SPRING, 2022) baseada em uma história de livreria. Spring é um *framework open source* para a plataforma Java, que torna fácil a criação de aplicações auto suficientes e robustas, possibilitando a execução imediata das mesmas. O projeto deveria obedecer algumas regras para o padrão organizacional da empresa, tais como: todos os nomes de variáveis, classes e métodos escritos em Inglês; criar *migrations* com uma ferramenta de migração de dados chamada flyway (FLYWAY, 2022) para os comandos SQLs; as pastas deveriam ser organizadas de acordo com as entidades, seguindo o modelo de padrão de projeto DTO; deve-se também colocar as classes de serviço em uma pasta interna a pasta da entidade, assim como serviços devem ser separados de acordo com a funcionalidade da qual o *controller* irá chamar; além de teste unitário individual para cada serviço. Foi solicitado que os serviços fossem separados em exagero, assim deixando o projeto melhor dividido em módulos.

Figura 01 - Modelo de padrão de projeto DTO para a entidade Book



Fonte: próprio autor

Figura 02 - Pasta v1 com o controller de livro



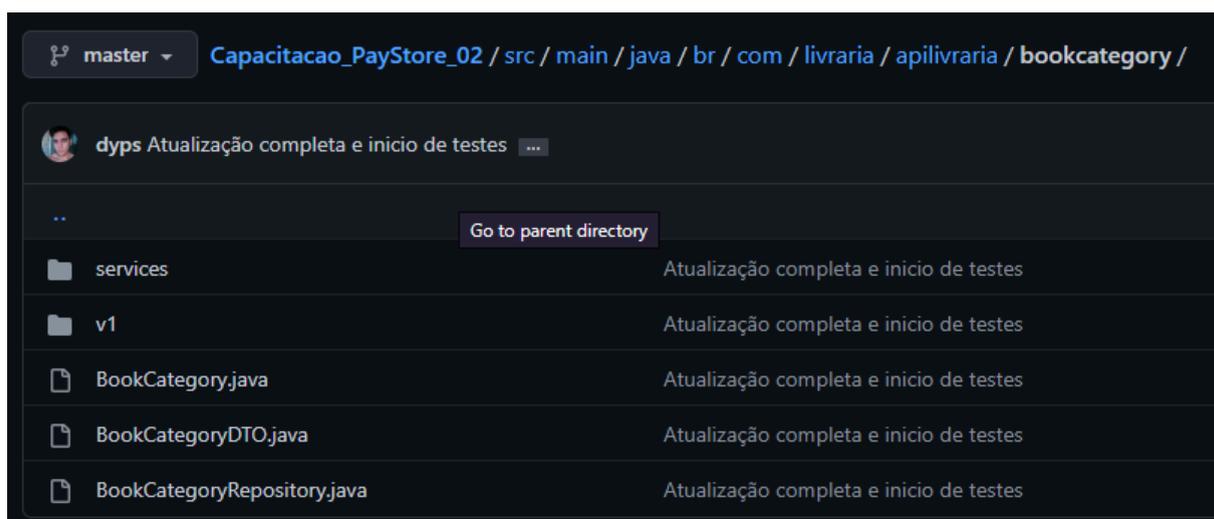
Fonte: próprio autor

Figura 03 - Pasta com os serviços da entidade livro



Fonte: próprio autor

Figura 04 - Entidade BookCategory



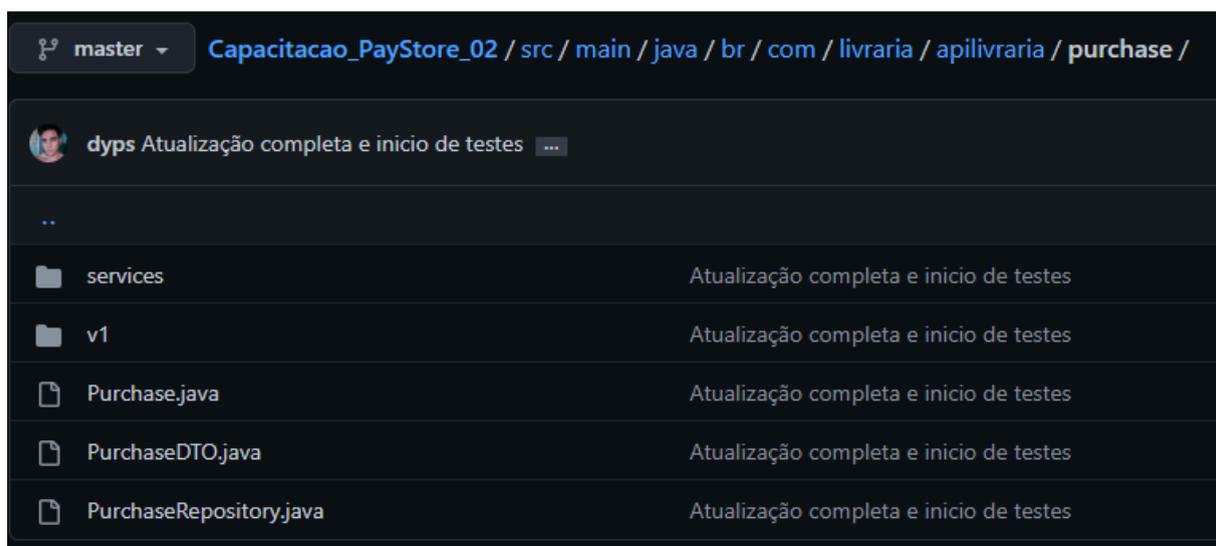
Fonte: próprio autor

Figura 05 - Entidade Client



Fonte: próprio autor

Figura 06 - Entidade Purchase



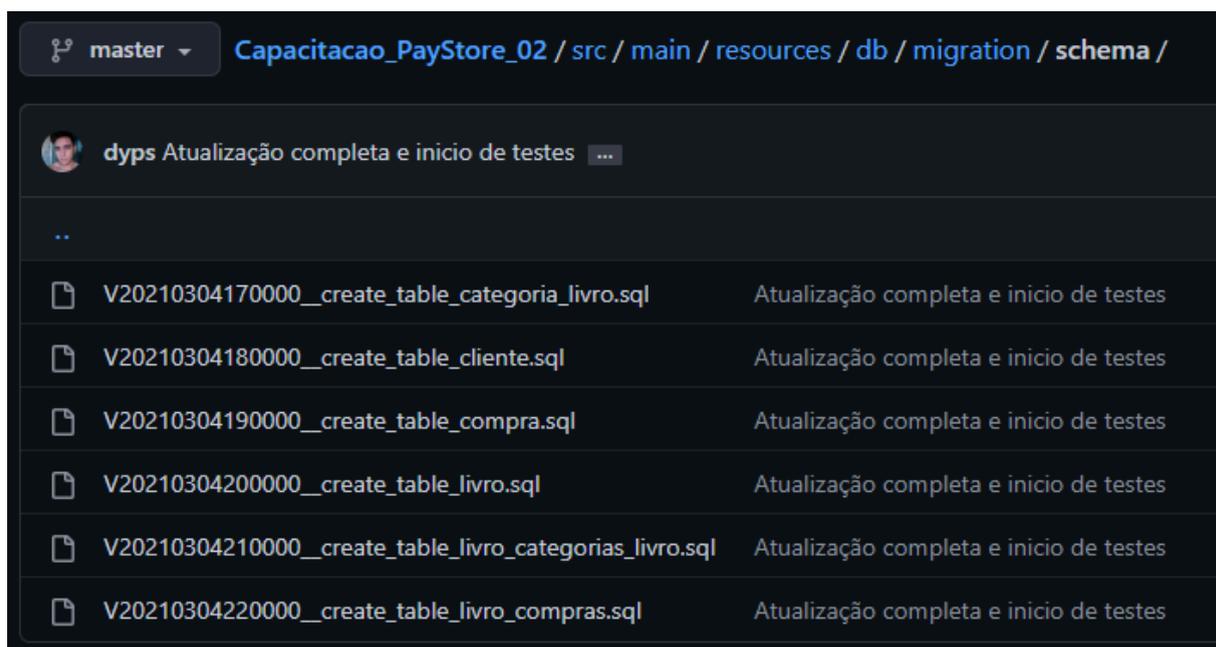
Fonte: próprio autor

Figura 07 - Pasta com as exceções da aplicação



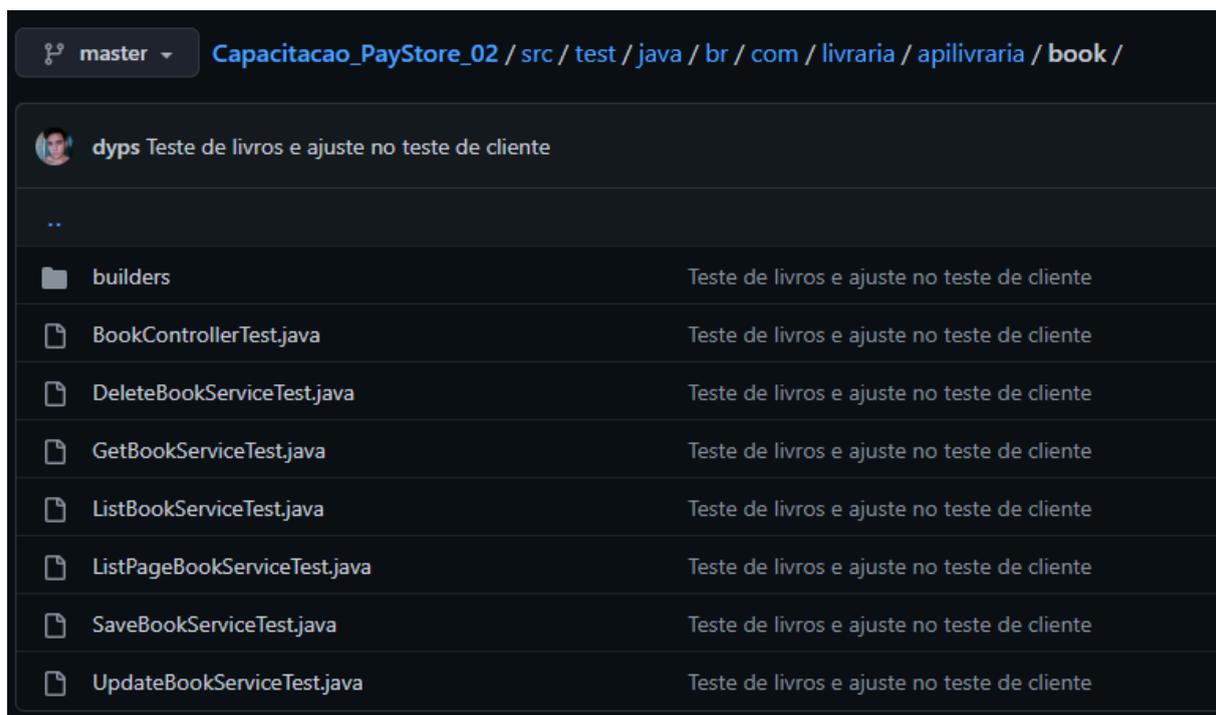
Fonte: próprio autor

Figura 08 - Pasta utilizada pelo flyway



Fonte: próprio autor

Figura 09 - Testes unitários da entidade Book



Fonte: próprio autor

Etapa 2: Introduzir o mundo dos microsserviços. Foi feita a separação do projeto, anteriormente monolítico, onde cada “entidade base” virou um novo projeto sendo totalmente apartado com suas próprias funcionalidades, *migrations* e comunicando-se com os outros quando necessário. Para isso, foi necessário um Eureka Server (EUREKASERVER, 2023), um projeto rodando com uma dependência que identifica e registra os clientes, assim direcionando as requisições para o cliente correto.

Figura 10 - Projetos em separado, com o server



Fonte: próprio autor

3.2.2 Desenvolvimento de funcionalidades do PayStore

PayStore (PAYSTORE, 2023) é um sistema desenvolvido pela empresa Phoebus que faz a captura do pagamento, armazena, dá instruções para as maquinetas de cartão e auxilia por completo aqueles que vendem as maquinetas.

Após esse período de capacitação, tarefas de implementação relacionadas ao PayStore passaram a ser alocadas para o estudante.

A empresa tem uma estrutura organizacional de mais de cento e cinquenta funcionários divididos por produtos. Na equipe PayStore, há a divisão em outras duas sub-equipes, a plataforma e a transacional. Na equipe da plataforma, da qual o autor fez parte, existe um líder, que também atua no desenvolvimento, e mais cinco desenvolvedores, um *Product Owner* (PO), três analistas de testes e duas pessoas responsáveis por criar os arquivos para as atualizações dos ambientes. É utilizada a metodologia Scrum (SUTHERLAND, 2014) para o gerenciamento do desenvolvimento das funcionalidades.

A empresa tem reuniões quinzenais para alocação de funcionalidades e divisão de tarefas em *releases*, e semanais para acompanhamento do desenvolvimento das mesmas. É utilizado para acompanhamento dessa gerência o Redmine (REDMINE, 2023), uma aplicação web de gerenciamento de projetos flexíveis, multiplataforma e *multidatabase*, onde são organizadas as tarefas que foram separadas nas histórias descritas no documento Drec, documento com descrições detalhadas elaboradas pelo PO.

Nas próximas seções, serão descritas as funcionalidades, na sequência em que foram alocadas, e como elas foram desenvolvidas pelo autor deste relatório.

3.2.3 Projeto PayStore - Release 2021.02

Nesta primeira *release*, foi dada uma tarefa para criar uma configuração para acesso e a documentação de todos os *endpoints* públicos do microserviço novo, *payment-storage*, que é responsável pelo armazenamento dos pagamentos já feitos. Usando *Swagger* (SWAGGER, 2023), *framework* que auxilia na modelagem, documentação e até geração de código para desenvolvimento de APIs, foi concluído em um pouco mais de uma semana com *scripts* que agilizam em muitos pontos basicamente lendo, identificando e escrevendo na documentação de acordo com o tipo. Um *bug* foi encontrado. foi observado que haviam formas de filtragem que se repetiam. Ele foi classificado como baixa prioridade e foi resolvido rapidamente enquanto a atividade anterior estava progredindo.

Outro bug foi encontrado na edição de uma entidade pelo portal. No campo nome da entidade não deveria ser possível colocar um valor já existente. Houve um pouco de dificuldade para a replicação do cenário por não estar bem especificado o fluxo. Após isso, a resolução foi simples e foi feito um cenário de teste unitário, utilizando o framework JUnit (JUNIT, 2023), específico para esse problema.

Um *bug-dev* foi encontrado na documentação da API externa do principal serviço a PS-API, API principal do PayStore, em um *endpoint* de verbo *post*. Era necessária a ida de um objeto que estava faltando na descrição. Foi uma correção rápida, assim que foi alinhada a maneira correta do objeto ser enviado.

Um *bug* foi encontrado no portal, o qual foi visto que um campo *booleano* sempre mostrava “não” na coluna, que referenciava e apresentava erro interno ao ser ordenado. A resolução foi uma alteração no DTO da entidade que aquela tela chama na API. Foram feitos testes manuais levantando três microsserviços, PS-API, PS-Webapp e PS-Proxy, para testar por completo.

Houve o cancelamento de um *bug* alocado para correção, pelo seguinte motivo: O *endpoint* utilizado durante o teste não condiz com o apresentado na documentação. O que nos remete a duas possibilidades: quando o bug foi cadastrado, ou a documentação estava errada na época, tendo em vista o tempo que o caso foi cadastrado, ou erro humano ao testar o *endpoint*.

3.2.4 Projeto PayStore - Release 2021.03

Um *bug-dev* foi encontrado em um *endpoint* de verbo *put* em um microsserviço chamado E-Commerce API (API criada para a execução das atividades de vendas online). O problema consistia em, ao enviar um campo, mesmo passando nulo, retornar um erro 500. A resolução foi ajustar o parse de uma biblioteca interna chamada *web-core* e subir a versão da mesma no microsserviço.

Nessa release, o gestor do time fez uma reunião para conversar sobre um comportamento estranho que acontecia em uma tela do portal que era responsável por salvar os dados da conta e testar o envio de email da aplicação. Tal atividade não era prioritária, porém, foi dado início a análise, pois somente seria implantada na próxima release.

3.2.5 Projeto PayStore - Release 2021.04

Uma melhoria foi criada, a partir da análise feita anteriormente. Foi visto que, ao testar o envio do email, sempre aparecia a mensagem de sucesso, o envio era feito de maneira assíncrona por uma biblioteca interna chamada Courier. A mesma estava dentro de outra biblioteca interna, Web-Core, já citada acima. Foi ajustado também para que não houvesse a necessidade de colocar a senha para testar o envio toda vez que abrisse a página, assim deixando necessária a digitação da senha somente quando houvesse mudança da mesma. Foi ajustada uma limitação no tamanho do erro que é mostrado ao usuário em uma biblioteca *frontend* por não termos total controle sobre o serviço de envio de email, por ser externo. Essa melhoria rendeu muitas alterações, tendo sido alterados um total de sete aplicações (Courier, Web-Core, Integrations, PS-Agent-Server, PS-Webapp, PS-API, Phrame-Angular).

Figura 11 - Tela do portal para controle do servidor de email

The screenshot shows a web interface for configuring an email server. The title bar is blue and contains a menu icon, the text 'Configurações > Servidor de Email', and a user profile icon 'G'. The main content area is divided into two panels: 'Configurações' and 'Segurança e Autenticação'. The 'Configurações' panel includes fields for 'Servidor*' (http://[redacted].cc), 'Porta*' (8801), 'Nome do Remetente*' ([redacted] - Certificação), 'Email do Remetente*' ([redacted]@[redacted].com.br), and 'Endereço da página*' (https://[redacted].com.br/). The 'Segurança e Autenticação' panel has toggle switches for 'Usar SSL' (off) and 'Usar TLS' (on), and input fields for 'Usuário*' ([redacted]@[redacted].com) and 'Senha' (0 / 255). At the bottom right, there are two buttons: 'TESTAR ENVIO DE E-MAIL' (yellow) and 'APLICAR' (blue with a checkmark).

Fonte: própria do autor

Uma melhoria para facilitar a vida de um grande cliente foi pensada e a resolutiva foi a adição de um novo campo *booleano*, o *DTO* de retorno de um *endpoint*, que indicava se possuía ou não assinatura, juntamente com um ajuste na *view* que retorna esse valor. Esse campo agilizaria para que não fosse necessária a ida em outro *endpoint* para ver a assinatura, o que era necessário antes para verificar se tinha ou não.

Uma nova funcionalidade para a *PS-API* foi dividida e uma parte era fazer o endpoint para a remoção da entidade alvo. Um *bug-dev*, após a entrega do *endpoint* foi encontrado. Ao tentar remover uma dessas entidades pela *API* externa da *PS-API* foi apresentado erro 409. Pelo portal, mostrava uma mensagem explicando que havia dependências que impediam a remoção. A funcionalidade foi ajustada para mostrar uma mensagem detalhada com erro 400.

Decorrente da entrega comentada anteriormente, outros dois *bugs* foram corrigidos rapidamente. No primeiro, o formato de um campo estava diferente na *API* correta no portal. No segundo, a quantidade de caracteres de alguns campos foi redefinida, tanto na *API* quanto no site.

Um *bug* de produção surgiu e mostrava a utilização de um filtro no portal. Esse filtro continha um botão de seleção com o valor repetido. O ajuste feito foi na tradução para o espanhol, os campos da seleção eram parecidos, um era estorno e o outro anulação.

3.2.6 Projeto PayStore - Release 2021.05

Foi criada uma tarefa para adicionar ao *frontend* um botão. Ele seria responsável por abrir a nova tela baseado na habilitação, ou não, de uma variável de ambiente. Após a implementação, surgiu um *bug-dev* nessa tela aberta, um campo de data mostrava uma data a frente da atual. O ajuste foi dado na *API* para que mandasse no formato adequado, pois enviava a data correta sem a região. Antes, a tela assumia que era o valor padrão da linha de Greenwich e adicionava horas, deixando incoerente.

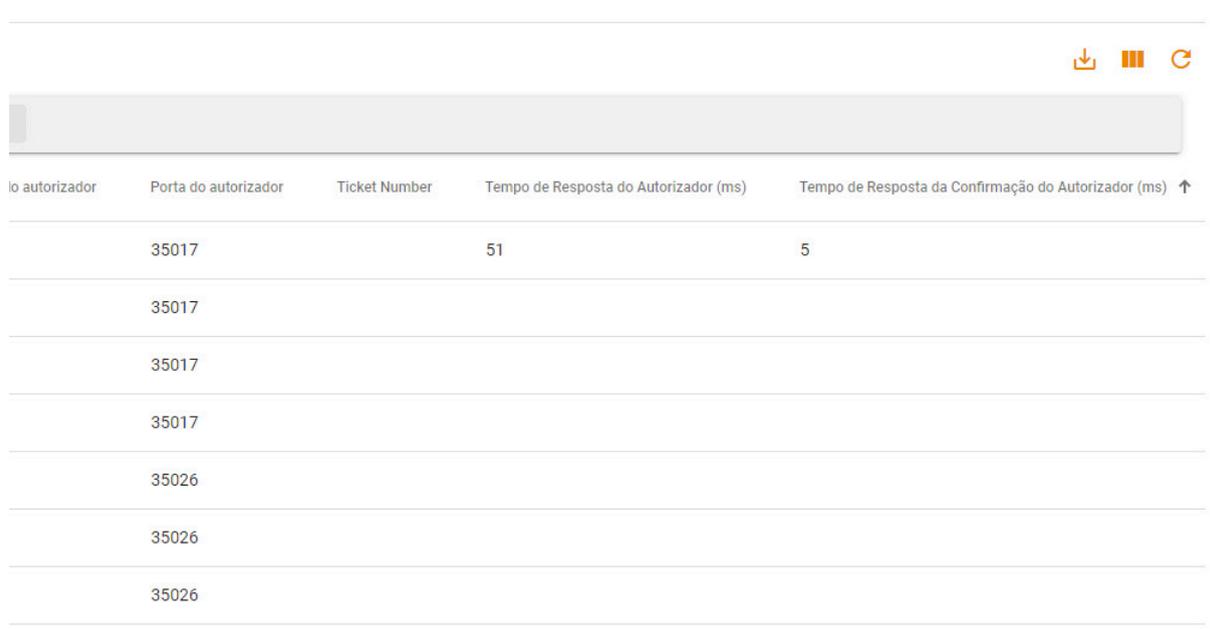
Um *bug* foi encontrado ao realizar uma consulta em um *endpoint* novo. O mesmo não diferenciava os *tokens* que nele chegavam, deixando para um usuário que possuía seu *token*, ver as informações de outros. Durante a análise, foi observado em outros pontos que também não ocorriam esse ajuste de contexto.

3.2.7 Projeto PayStore - Release 2021.06

Foi necessário um aumento na cobertura de testes em um serviço utilizado em uma melhoria. Em outra, foi necessária a adição de duas novas colunas na entidade

que era a referência para o tempo de resposta de um serviço externo e uma duas colunas na grid no site.

Figura 12 - Tela do portal com as novas colunas



The screenshot shows a table with the following columns: 'lo autorizador', 'Porta do autorizador', 'Ticket Number', 'Tempo de Resposta do Autorizador (ms)', and 'Tempo de Resposta da Confirmação do Autorizador (ms)'. The data rows are as follows:

lo autorizador	Porta do autorizador	Ticket Number	Tempo de Resposta do Autorizador (ms)	Tempo de Resposta da Confirmação do Autorizador (ms)
			51	5
		35017		
		35017		
		35017		
		35017		
		35026		
		35026		
		35026		

Fonte: própria do autor

Em uma outra melhoria foi necessário mudar um *endpoint* que exporta arquivos de pagamentos para a maquininha de cartão, na PS-Api, para retornar a nova versão do arquivo, que a nova versão que o terminal recebeu precisava. A escolha seria feita via *queryparam* na chamada da requisição.

3.2.8 Projeto PayStore - Release 2021.07

Diversas vezes um *bug* foi republicado, um desenvolvedor e um testador ficaram aprimorando e alinhando a melhor maneira de mostrar informações relevantes em uma tela nova, que faz as relações entre um terminal e cada um de seus *apps*. A princípio, aos olhos do cliente, era inútil ou continha informações incoerentes com suas descrições. Após essa resolução, a tela se tornou uma importante área para saber relatórios dos terminais para o cliente.

Foi corrigido um *bug* simples que não mostrava o valor de um campo, a correção foi um ajuste para preencher o campo no DTO da API.

3.2.9 Projeto PayStore - Release 2021.08

Foi criada a “base” de uma nova tela no serviço PS-Webapp, ela abriria internamente a tela do novo serviço, Mobitef-Webapp, serviço esse que mostraria

tabelas e gráficos dos pagamentos e demais recursos feitos, usando o aplicativo Mobitef, serviço que integra o comércio aos meios de pagamento, é um elo de comunicação entre automação e transações (MOBITEF, 2023), presente nas maquininhas. Também foram criadas permissões para o acesso a essa tela.

Um *bug* por inconsistência em uma *procedure* no serviço PS-API foi corrigido e de imediato foi realizada a implantação de um novo *patch* em produção.

3.2.10 Projeto PayStore - Release 2021.09

Uma melhoria para desabilitar o uso da ferramenta Kafka foi feita. Um cliente que não utilizava, solicitou a remoção, pois como ficava tentando conectar, acabava sujando os *logs* de erros, o que poderia ofuscar um erro mais grave. Apache Kafka é uma plataforma distribuída de transmissão de dados que é capaz de publicar, subscrever, armazenar e processar fluxos de registro em tempo real (REDHAT, 2023).

Uma nova tela foi criada no serviço Onboarding-Webapp, serviço esse responsável por telas mais modernas implementadas em Angular. Foi solicitada a criação do botão de remoção e outro bem mais complexo que necessitou da criação completa de sua arquitetura na biblioteca interna Ngx-Phoebus-Table.

3.2.11 Projeto PayStore - Release 2021.10

Foi feita uma implementação de um novo serviço chamado *onboarding-api*, serviço esse feito para desacoplar alguns cadastros e também conseguir suprir melhor grandes demandas. O PO fez um DREC para explicar a demanda. Durante a leitura, foram reparados alguns pontos interessantes de melhoria e de ajuste. A partir daí, foi dado o poder de repensar a missão da criação das entidades básicas. Foi também feito uma melhoria na lógica do negócio da documentação, desde antes do início da implementação e durante a medida do necessário. A partir dessa demanda, foi implementado um *endpoint* de listagem e também foram resolvidos outros cinco *bugs* simples, de tipo de valor do atributo diferente do da especificação, a falta de um erro melhor descrito ao passar valor indevido em um campo, um retorno de um status não documentado que foi basicamente uma verificação se está no status de controle interno, a remoção de informações desnecessárias e uma melhor organização do local de alguns atributos.

3.2.12 Projeto PayStore - Release 2021.11

Foi dado início a uma transição de parte essencial do projeto PS-Api, que utiliza banco *Oracle* (ORACLE, 2022), para um microsserviço chamado Payment-Storage, para um que usa *Mongo* (MONGO, 2022), a fim de tirar parte da carga do *Oracle* e usar um banco que aguenta mais carga e tenha resultados mais rápidos. Nessa *release*, foram resolvidos cinco *bugs* relacionados a essa iniciativa. Neles continham vários campos divergentes entre os *endpoints* que deveriam ser os substitutos equivalentes, ou seja, deveriam dar o mesmo resultado, ou com mudanças esperadas para casos de melhorias no antigo fluxo. Os *bugs* encontrados nos *endpoints* legados foram descartados e pontuados em planilhas para todos os integrantes da equipe.

3.2.13 Projeto PayStore - Release 2021.13

Foi criada uma entidade nova que visava alterar o fluxo de um equipamento, pois todos aqueles que tivessem naquela entidade deixariam de seguir alguns *endpoint* e perderíamos algumas referências de valores para que houvesse espaço para uma nova forma de gerência desses equipamentos. Também foram alterados alguns *endpoints* antigos para ter sua funcionalidade alterada de acordo com a existência, ou não, dessa nova entidade, apontando para aquele equipamento. Foram alteradas *procedures* do negócio para executar, ou não, também de acordo com essa entidade naquele equipamento.

Foi criado um novo tópico de consumo no Kafka já que alguns valores necessários para a criação dessa nova entidade vinham pelo mesmo.

Também foi feita a chamada de um *put* em um outro microsserviço, de um outro *squad*, para manter a integridade das aplicações, a partir da entidade criada anteriormente.

Foi necessária a criação de 15 índices de múltiplos valores a partir de uma análise feita no banco de produção por uma equipe externa especializada. Eles foram necessários para que fosse possível acessar uma tela que não carregava de maneira nenhuma pela grande quantidade de dados da mesma.

3.2.15 Projeto PayStore - Release 2021.14

Foi criado um mecanismo de sincronização do microserviço App-Manager, serviço de gerência dos aplicativos das maquininhas de cartão, do outro squad, a partir de uma variável de ambiente na PS-API que, ao ser iniciado o mesmo de maneira assíncrona, faz chamadas aos *endpoints*, conferindo de maneira inteligente o que já está salvo e adicionando o que deveria estar.

3.2.16 Projeto PayStore - Release 2021.15

Foi criado um *endpoint* na PS-API que manda para a tela um *link* único para que possa enviar um arquivo csv a um servidor Minio, servidor de armazenamento de objetos, distribuído de alto desempenho, que é projetado para infraestrutura de nuvem privada em larga escala (MINIO, 2023). A partir do arquivo, foi criado um *endpoint* interno que o processa e salva no banco os dados nele contido. Esse endpoint é chamado de tempos em tempos por um microserviço Ps-Agent que realiza operações.

3.2.18 Projeto PayStore - Release 2022.01

Foi feita uma tarefa para a adição de uma nova configuração visual dos dispositivos que, a partir de uma versão x do aplicativo de sistema, terá nova aparência deixando a antiga ainda funcional. Foi ajustado para que o *endpoint* que indica as operações que o dispositivo deve fazer avise sobre o visual novo.

Foi feito um estudo quanto a viabilidade e as novas propostas para um produto interno que foi criado para validar dados de múltiplos *endpoints* e comparar os resultados, apresentando somente pontos de divergência. O problema principal consistia na inviabilidade decorrente da demora no processamento e do resultado final, pois em caso de uso, em um dia estaria chegando na casa dos milhões de resultados.

3.2.19 Projeto PayStore - Release 2022.02

Foi feito um trabalho de levantamento de bibliotecas internas usadas em todos os microserviços de *frontend*.

Foi feita a integração a um serviço externo de autenticação de usuários e ajustes para a utilização do novo tipo de *token* gerado.

4 CONSIDERAÇÕES FINAIS

A realização do estágio foi de grande importância para o autor, pois, durante o ano descrito na empresa, foi possível aprimorar conhecimentos, subir de cargo, melhorar a maneira de pensar a nível de negócio. Foram momentos de muito aprendizado, momentos onde pude perceber que o escalonamento pode começar a afetar o código e a otimização completa é trivial.

As atividades que foram alocadas, ao longo deste ano, foram cumpridas e também feitos dados diversos elogios, por saber reconhecer corretamente os *bugs* e por trabalhar de maneira inteligente e eficiente. Foram utilizadas técnicas aprendidas nas disciplinas de análise de algoritmos, análise de testes e desenvolvimento de aplicações corporativas, além de princípios aprendidos nas disciplinas de estrutura de dados e algoritmos, análise de algoritmos, padrões de projeto. Também, durante o estágio, foram apreendidas tecnologias novas como Swagger (SWAGGER, 2023) e Minio (MINIO, 2023). Com isso, o estágio foi cumprido de forma satisfatória, tendo sido reconhecido o trabalho na empresa, estando neste momento com contrato vigente de trabalho.

REFERÊNCIAS

ANGULAR. Delivery Web Apps with confidence. Disponível em: <https://angular.io>. Acesso em: 08 set. de 2022.

EUREKASERVER. Eureka Service Discovery. Disponível em: <https://cursos.alura.com.br/forum/topico-eureka-service-discovery-124159>. Acesso em: 03 mar. de 2023.

FLYWAY. Increase reliability of deployments by versioning your database. Disponível em: <https://flywaydb.org>. Acesso em: 08 set. de 2022.

JUNIT. JUnit 5. Disponível em: <https://junit.org/>. Acesso em: 24 fev. de 2023.

MINIO. Multi-Cloud Object Storage. Disponível em: <https://min.io>. Acesso em: 24 fev. de 2023.

MOBITEF. Prezi. Disponível em: https://prezi.com/p/rod5_gck_8vr/mobitef/. Acesso em: 03 mar. de 2023.

MONGO. Build the next big thing. Disponível em: <https://www.mongodb.com>. Acesso em: 08 set. de 2022.

ORACLE. Oracle Brasil. Disponível em: <https://www.oracle.com/br/>. Acesso em: 08 set. de 2022.

PayStore. PayStore | Seja mais que uma empresa de pagamentos! Disponível em: <https://www.paystore.com.br/>. Acesso em: 03 mar. de 2023.

PHOEBUS. Tecnologia em Pagamentos. Disponível em: <http://www.phoebus.com.br>. Acesso em: 08 set. de 2022.

REDHAT. Apache Kafka is a distributed data streaming platform that can publish, subscribe to, store, and process streams of records in real time. It is designed to handle data streams from multiple sources and deliver them to multiple consumers. Disponível em: <https://www.redhat.com/pt-br/topics/integration/what-is-apache-kafka>. Acesso em: 24 fev. de 2023.

REDMINE. Overview – Redmine. Disponível em: <https://www.redmine.org>. Acesso em: 24 fev. de 2023.

SPRING. Why Spring? Spring, 2023. Disponível em: <https://spring.io/why-spring/>. Acesso em: 09 jan. de 2023.

SWAGGER. API Development for Everyone. Disponível em: <https://swagger.io>. Acesso em: 24 fev. de 2023.

SUTHERLAND, Jeff; SUTHERLAND, J.J. SCRUM : a arte de fazer o dobro do trabalho na metade do tempo. São Paulo: Leya, 2014. E-book (288 p.). ISBN: 978-85-431-0717-2.

GLOSSÁRIO

Csv: Tipo de arquivo usado em planilhas

Endpoint: “Ponto de extremidade”, ele é um link ou caminho para conectar a um servidor.

Bug-Dev: Problema com a aplicação encontrado em primeira instância, ou seja durante o período de testes internos.

Bug: Problema com a aplicação encontrado após uma versão já disponível para o cliente.

Api: Aplicação com diversos endpoints.

Frontend: Parte da aplicação que é visível ao usuário.

Microsserviços: Diversas aplicações que juntas formam o domínio do negócio.

PO: Product Owner, pessoa que representa os interesses de todos os envolvidos.

Drec: Documento com descrições detalhadas elaboradas pelo PO.

Logs: Mensagens da aplicação durante o seu uso, nele mostra as linhas que “explodem” quando há algum erro.

DTO: Entidade que imita a entidade real, ela é usada por segurança para que só veja ou altere o que é permitido.

Procedure: Estrutura de banco de dados que realiza procedimentos.

View: Estrutura de banco de dados que dá uma visão mais especializada.

Put: Método usado em endpoints, ele é usado principalmente para atualizações.

Queryparam: Parâmetro de um endpoint, ele pode conter valores a serem utilizados na aplicação, eles podem ser ou não obrigatórios.

Booleano: Tipo de valor de variável que pode ter dois valores: verdadeiro ou falso.

Squad: Grupo de pessoas

Minio: É um armazenamento de objetos de alto desempenho

	INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA
	Campus Monteiro - Código INEP: 25284940
	Pb-264, S/N, Serrote, CEP 58500-000, Monteiro (PB)
	CNPJ: 10.783.898/0008-41 - Telefone: (83) 3351-3700

Documento Digitalizado Restrito

Relatório de Estágio

Assunto:	Relatório de Estágio
Assinado por:	Yaggo Pereira
Tipo do Documento:	Anexo
Situação:	Finalizado
Nível de Acesso:	Restrito
Hipótese Legal:	Informação Pessoal (Art. 31 da Lei no 12.527/2011)
Tipo do Conferência:	Cópia Simples

Documento assinado eletronicamente por:

- Dayvid Yaggo Pereira Souza, ALUNO (201615020160) DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS - MONTEIRO, em 30/04/2025 15:43:32.

Este documento foi armazenado no SUAP em 30/04/2025. Para comprovar sua integridade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifpb.edu.br/verificar-documento-externo/> e forneça os dados abaixo:

Código Verificador: 1477094

Código de Autenticação: 1eead40f25

